

**Progettazione di Applicazioni Web T**  
**Prova d'Esame – 29 giugno 2021**

**Tempo a disposizione: 120 minuti**

---

La soluzione comprende la consegna elettronica dei seguenti file:

<b>Pony.zip</b>	file zip contenente il sorgente java/class, file XML e txt per il punto 1
<b>P2018.zip</b>	file zip contenente il sorgente java/class, file XML e txt per il punto 2
<b>WebRatio.zip</b>	file zip contenente il testo di risposta (txt/doc/pdf) al punto 3

**Ogni file .zip consegnato DEVE CONTENERE TUTTI e SOLI i file creati/modificati e/o ritenuti importanti in generale ai fini della valutazione (ad esempio, codice java e relativi .class, descrittori XML, file txt/doc/pdf, ecc.) e NON dell'intero progetto.**

**N.B. La prova si intende superata se il punteggio finale non è inferiore a 18 punti (su un totale di 33). In particolare, è necessario (ma non sufficiente) ottenere la sufficienza in tutti gli esercizi.**

---

**ESERCIZIO 1 - 14 punti (sufficienza: 7 punti)**

Si progetti una grammatica **XML Schema**, e un **documento XML** di esempio, per la modellazione delle informazioni relative alla **Ludoteca “Pony”** nel rispetto delle seguenti specifiche:

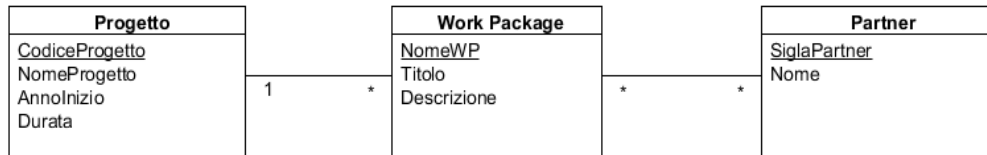
- I giocattoli si dividono in due categorie: giochi in formato digitale (“GD”), quali audiovisivi (“AV”) e cd musicali (“CDM”), e giochi tradizionali (“GT”) basati sull’uso e il tatto, quali, ad esempio, peluche, cavallo a dondolo, costruzioni lego, puzzle, giochi di società, ecc.
- A fini di ricerca nel database, ogni giocattolo è caratterizzato da un codice univoco obbligatorio.
- Inoltre, (i) ogni audiovisivo/cd musicale ha associato informazioni obbligatorie quali il titolo, l’autore, la casa produttrice, l’anno di produzione, la fascia di età (es. 0- 3, 4-6, ecc.), la lingua (es., “italiano”, “inglese”, ecc. ) e la durata in minuti; (ii) ogni gioco tradizionale ha invece associato informazioni obbligatorie quali il nome, l’autore, la marca, la fascia di età (es. 0+, 1-2, 3-5, 6-8, ecc.) e la modalità di utilizzo, ovvero gioco individuale (“GI”) oppure gioco di gruppo (“GG”).

Si realizzi quindi l’applicazione **Java “Pony”** che, facendo uso del **parser SAX** e del **documento XML di esempio** realizzato al punto precedente, esponga il metodo `getGiocoTradizionale_Fascia6-8_GI()`, unitamente a un suo `main` di prova in grado di restituire il nome e la marca dei giochi tradizionali, di fascia 6-8 e di modalità individuale. Stampare il risultato opportunamente formattato prodotto dal metodo `getGiocoTradizionale_Fascia6-8_GI()` sul file **Pony.txt**.

**Progettazione di Applicazioni Web T**  
**Prova d'Esame – 29 giugno 2021**

**ESERCIZIO 2 - 14 punti (sufficienza: 7 punti)**

Partendo dalla realtà illustrata nel diagramma UML di seguito riportato, si fornisca una soluzione alla gestione della persistenza basata su **Hibernate** in grado di “mappare” efficientemente e con uso di ID surrogati il modello di dominio rappresentato dai **JavaBean** “Progetto”, “Work Package” e “Partner” del diagramma UML con le corrispondenti **tabelle relazionali derivate dalla progettazione logica del diagramma** stesso.



Nel dettaglio, dopo aver creato da applicazione Java le tabelle all'interno del proprio **schema** nel database **TW\_STUD** di **DB2** (esplicitando tutti i **vincoli** opportuni di PK e FK), implementato i **JavaBean**, definiti i **file XML di mapping** e il **file XML di properties**, si richiede la realizzazione di una classe di prova facente uso delle **API Hibernate** in grado di:

- **istanziare** alcuni JavaBean “Progetto”, “Work Package” e “Partner”, rendendoli persistenti rispetto alla base di dati associata al diagramma UML;
- **selezionare:** (i) per i progetti iniziati nell’anno “2018”, il nome del progetto, il numero di work package di cui si costituisce e, per ognuno di questi, il numero di partner che vi lavorano; (ii) nome del partner che partecipa a più work package; **producendo una stampa opportunamente formattata del risultato** sul file **Progetti-2018.txt**;
- **eliminare** i progetti restituiti al punto precedente;

il tutto mediante opportuna demarcazione delle transazioni.

**N.B.** L’implementazione **deve limitarsi** al solo **DBMS DB2**. La soluzione Java **deve sfruttare esplicitamente i mapping N-M e 1-N/N-1** specificati nell’UML. Ogni ulteriore scelta da parte dello studente deve essere opportunamente giustificata con commenti nel codice.

**ESERCIZIO 3 - 5 punti (sufficienza: 3 punti)**

Considerando l’ipertesto progettato negli **schemi IFML di WebRatio** ai punti **a), b), c) e d)** di seguito riportati, il candidato descriva le tipologie di link esistenti tra i componenti pagina e lista **pA**, **pB**, **IA** e **IB**.

